

# Optimasi K-Nearest Neighbor Dengan Particle Swarm Optimization Untuk Memprediksi Harga Komoditi Karet

Rudy Donny Liklikwatil<sup>1</sup>, Edi Noersasongko<sup>2</sup>, Catur Supriyanto<sup>3</sup>

<sup>1</sup>STMIK Dipanegara, <sup>2,3</sup> Universitas Dian Nuswantoro

<sup>1</sup>[rudydonny@gmail.com](mailto:rudydonny@gmail.com), <sup>2</sup>[edi-nur@dsn.dinus.ac.id](mailto:edi-nur@dsn.dinus.ac.id), <sup>3</sup>[catursupriyanto@dsn.dinus.ac.id](mailto:catursupriyanto@dsn.dinus.ac.id)

## Abstrak

Komoditi dan harganya karet mengalami perubahan yang fluktuatif dan menunjukkan pola yang tidak stasioner, di sisi lain pengambilan keputusan bisnis memerlukan data yang akurat dan terukur. Algoritma k-NN merupakan algoritma yang merupakan algoritma unsupervised, dan terbukti baik pada data mining. Sedangkan Particle Swarm Optimization (PSO) menunjukkan performa optimasi yang lebih baik dibandingkan dengan metode yang lain. Penelitian ini bertujuan untuk merancang metode prakiraan yang dapat memperkirakan tingkat harga dan volume permintaan untuk TSR 20. Prakiraan dilakukan menggunakan Jaringan Syaraf Tiruan dengan algoritma propagasi balik, dimana data yang digunakan adalah data perkembangan pasar TSR pada bursa berjangka SICOM.. Berdasarkan tiga indikator pelatihan yang dijadikan acuan dalam pemilihan arsitektur terbaik, spesifikasi ke 15 tidak perlu melakukan pelatihan sampai epoch maksimum. Dalam teknik PSO terdapat beberapa cara untuk melakukan pengoptimasian diantaranya meningkatkan bobot atribut terhadap semua atribut atau variable yang dipakai, menseleksi atribut dan fitur seleksi. Hasil penelitian dari prediksi harga komoditi karet dengan menggunakan model k-NN mendapatkan nilai RMSE sebesar 0,087 sedangkan bila menggunakan k-NN yang dioptimasi dengan menggunakan Particle Swarm Optimization didapatkan nilai RMSE sebesar 0,082 lebih baik dibandingkan dengan hanya menggunakan k-NN saja.

**Kata kunci :** Prediksi Harga Karet, k-NN, Particle Swarm Optimization

## Abstract

*Commodities and prices of rubber undergo fluctuating changes and show patterns that are not stationary, on the other hand business decision making requires accurate and measurable data. The k-NN algorithm is an algorithm that is an unsupervised algorithm, and is well proven in data mining. While Particle Swarm Optimization (PSO) shows better optimization performance compared to other methods. This study aims to design a forecasting method that can estimate the price level and volume of demand for TSR 20. Forecasts are made using Artificial Neural Networks with a back propagation algorithm, where the data used is data on TSR market development on the SICOM futures exchange .. Based on three training indicators referenced in the best architectural selection, the 15th specification does not need to do training until the maximum epoch. In the PSO technique there are several ways to optimize, including increasing the weight of attributes to all attributes or variables used, selecting attributes and selection features. The results of the prediction of the price of rubber commodities using the k-NN model obtained an RMSE value of 0.087 while when using k-NN which was optimized using Particle Swarm Optimization the RMSE value was 0.082 better than just using k-NN.*

**Keywords:** Rubber Price Prediction, k-NN, Particle Swarm Optimization

---

## 1. PENDAHULUAN

Karet alam merupakan komoditas perkebunan yang memiliki peran cukup strategis dalam perekonomian bangsa. Dalam neraca perdagangan Negara tercatat bahwa karet merupakan penyumbang devisa Negara kedua terbesar setelah kelapa sawit. Karet merupakan komoditi yang tingkat permintaan dan produksinya sering mengalami peningkatan yang signifikan setiap waktu. Hal ini diakibatkan karena tingginya permintaan dari perusahaan pemasok hasil dari olahan karet untuk memenuhi kebutuhan produksinya. Komoditi karet sangat dikenal dan sudah dibudidayakan dalam jangka waktu yang lama dibandingkan dengan beberapa komoditi lainnya. Agroindustri karet ini memiliki peran yang penting karena bisa menjadi sumber devisa Negara, sumber lapangan kerja dan sebagai pemasok dari bahan baku industri barang jadi yang terbuat dari karet seperti ban. Karet merupakan salah satu komoditi penting di Indonesia.

Pasar domestik dan internasional semakin terbuka dan persaingan pasar tidak terhelakkan lagi sebagai konsekuensi logis akan menjadi semakin meningkat. Karena semua negara diharuskan melakukan pembukaan akses pasar dan penghapusan tarif dan subsidi. Sektor pertanian tidak dikecualikan dalam agenda perdagangan bebas tersebut [1], sehingga data harga komoditas mengalami perubahan yang fluktuatif dan menunjukkan pola yang tidak stasioner, oleh karena itu diperlukan suatu metode untuk melakukan prediksi harga suatu komoditi guna sebagai bahan pertimbangan bagi pengusaha ataupun bagi pihak-pihak yang berkepentingan untuk mengambil keputusan atau tindakan guna kelangsungan bisnis dan ekonomi.

Pada umumnya, prediksi menggunakan metode *time series* adalah cara untuk mengetahui sebuah nilai dimasa yang akan datang, dimana pengamatannya dilakukan berdasarkan urutan waktu. Proses determinasi dari metode *time series* memiliki kemampuan prediksi yang tinggi, nilai yang akan datang dapat diketahui dengan melihat nilai dari masa lalu [2].

Salah satu algoritma yang dapat memprediksi adalah *k-Nearest Neighbor* (k-NN). Metode k-NN merupakan salah satu algoritma *Machine Learning* (ML) yang dianggap sebagai suatu metode yang sederhana untuk diterapkan dalam analisis data dengan dimensi perubah yang banyak. Walaupun metode ini sederhana namun metode ini memiliki kelebihan dibandingkan metode lain, yaitu dapat menggeneralisasi himpunan data *training* yang relatif kecil. Pada awalnya k-NN merupakan metode untuk analisis klasifikasi, namun beberapa dekade terakhir digunakan untuk prediksi [3].

Khalib Alkatib[4], et al, melakukan penelitian untuk memprediksi harga saham dengan menggunakan 5 sampel perusahaan besar yang terdaftar di bursa Yordania dengan menggunakan algoritma *k-Nearest Neighbor* yang mempunyai kelebihan antara lain ketangguhan terhadap *training data* yang memiliki banyak *noise* dan efektif apabila training datanya besar[5], diperlukan waktu yang cepat dalam *training*[6] dan algoritma ini mampu memberikan hasil rasio kesalahan yang kecil untuk prediksi[4].

Pada penelitian yang dilakukan oleh Atthirawong. W, et al [5] menggunakan *Neural Network* untuk prediksi harga karet ini, ada dua model jaringan saraf yang diusulkan untuk memprediksi harga karet Hatyai pasar karet sentral di Thailand dan membandingkan akurasi kinerja. Model pertama adalah model A yaitu model *single time series*. Dalam model ini, hanya ada data harian harga karet selama 2001-2003 digunakan sebagai data masukan untuk meramalkan nilai-nilai masa depan. Model kedua adalah model *time series* multivariat disebut Model B. Kesalahan Model B sedikit lebih kecil dibandingkan Model A. Hal ini tersirat bahwa Model B mencapai kinerja yang lebih tinggi dibandingkan Model A. Dengan kata lain, dapat direkomendasikan bahwa nilai tukar mata uang yen bisa membantu meningkatkan akurasi dalam memprediksi kinerja harga karet. Model A mendapat RMSE 0,5295 dan MAPE 1,2498 sedangkan Model B mendapat RMSE 0,4661 dan MAPE 0,9961. Adapun kelemahan dari penelitian ini adalah untuk meningkatkan akurasi prediksi, salah satu cara adalah memperhitungkan penyertaan informasi lainnya misalnya harga karet alam di berbagai pasar di seluruh dunia dan faktor geografis dalam model, selain itu data *time series* yang lebih lama dapat meningkatkan hasil prediksi.

Pada penelitian yang dilakukan oleh Nofi Erni, *et al.*[6] bertujuan untuk merancang suatu metode prakiraan yang dapat memperkirakan tingkat harga dan volume permintaan untuk TSR 20. Prakiraan dilakukan menggunakan Jaringan Syaraf Tiruan (JST) dengan algoritma propagasi balik, dimana data yang digunakan adalah data perkembangan pasar TSR pada bursa berjangka SICOM.. Berdasarkan tiga indikator pelatihan yang dijadikan acuan dalam pemilihan arsitektur terbaik, spesifikasi ke 15 tidak perlu melakukan pelatihan sampai *epoch* maksimum. Pada *epoch* ke 50 target (*goal*) sudah tercapai dengan nilai MSE mencapai 0,0000919 dan nilai koefisien regresi mencapai 0,9977

Adapun kelemahan dari k-NN yaitu menggunakan seluruh data *training* untuk proses klasifikasi (*complete storage*) yang dapat mengakibatkan proses prediksi yang lama [6]. Untuk mengatasi kelemahan algoritma k-NN digunakan metode pemilihan fitur (*feature selection*). *Particle Swarm Optimization* (PSO) digunakan untuk memecahkan masalah optimasi yang disertai dengan berbagai fitur. Dalam teknik *Particle Swarm Optimization* (PSO) terdapat beberapa cara untuk melakukan pengoptimasian diantaranya meningkatkan bobot atribut (*attribute weight*) terhadap semua atribut atau variable yang dipakai, menseleksi atribut (*attribute selection*) dan fitur seleksi (*feature selection*) [7].

Berdasarkan uraian diatas, metode yang digunakan dalam penelitian ini adalah algoritma *k-Nearest Neighbor* (*k-NN*) berbasis *Particle Swarm Optimization* (PSO) untuk prediksi time series harga komoditi karet spesifikasi teknis.

### 1.1 Prediksi

Prediksi adalah memperkirakan sesuatu yang akan datang. Kejadian masa lampau dipelajari untuk menentukan kecenderungan pola datanya. Jenis prediksi berdasarkan horizon perencanaan. Terdiri dari (1) Prediksi jangka pendek, jika waktu prediksi kurang dari 3 bulan, (2) Prediksi jangka menengah, jika jangka waktu prediksinya tiga bulan hingga dua tahun, (3) Prediksi jangka panjang, jika jangka waktu lebih dari dua tahun. Sedangkan teknik prediksi dapat dikategorikan menjadi (1) *Time Series*, yang termasuk kategori ini adalah *moving average*, *exponensial smooting*, *box Jenkins* (2) *Causal*, hasil keluaran dipengaruhi secara langsung oleh beberapa factor lain. Yang termasuk kategori ini adalah *regression model*, *judgemental*, *experopinion*, *Delpi*, *simulation*[8].

Berdasarkan sifatnya, peramalan dibedakan menjadi:

1. Peramalan Kualitatif, Peramalan yang didasarkan atas data kualitatif pada masa lalu. Hasil peramalan kualitatif didasarkan pada pengamatan kejadian-kejadian di masa sebelumnya digabung dengan pemikiran dari penyusunnya.
2. Peramalan Kuantitatif, Peramalan yang didasarkan atas data kuantitatif masa lalu yang diperoleh dari pengamatan nilai-nilai sebelumnya. Hasil peramalan yang dibuat tergantung pada metode yang digunakan, menggunakan metode yang berbeda akan diperoleh hasil peramalan yang berbeda[9].

### 1.2 Times Series

*Time series* adalah himpunan nilai-nilai hasil pengamatan (x)<sub>t</sub> berdasarkan periode waktu dan disusun untuk melihat pengaruh perubahan dalam rentang waktu tertentu[16]. Prediksi data *Time Series* adalah pendugaan data yang akan datang yang dilakukan berdasarkan nilai data masa lalu dari suatu variable[10].

Saat ini analisis data *Time series* telah menjadi salah satu cabang matematika statistik yang penting dan luas digunakan dalam berbagai bidang keilmuan. Analisis data *Time series* dapat digunakan untuk mengetahui perkembangan suatu atau beberapa kejadian dan hubungan atau pengaruhnya pada kejadian lainnya, sekaligus menghitung secara kuantitatif besarnya pengaruh suatu kejadian dengan kejadian lainnya apabila berhubungan. Data *Time series* juga dapat digunakan untuk membuat ramalan-ramalan berdasarkan garis regresi atau *trend*[11].

### 1.3 Particle Swarm Optimization

*Particle Swarm Optimization* (PSO) digunakan untuk memecahkan masalah optimasi yang disertai dengan berbagai fitur. Dalam teknik *Particle Swarm Optimization* (PSO) terdapat beberapa cara untuk melakukan pengoptimasian diantaranya meningkatkan bobot atribut (*attribute weight*) terhadap semua atribut atau variable yang dipakai, menseleksi atribut (*attribute selection*) dan fitur seleksi (*feature selection*) [7].

Sebuah partikel diimplementasikan sebagai sebuah *Vector* dengan posisi sebuah partikel ke- $i$  di representasikan sebagai  $x_i = x_{i1}, x_{i2}, x_{i3}, \dots, x_{id}$  dan kecepatan partikel ke- $i$  dipresentasikan sebagai  $v_i = v_{i1}, v_{i2}, v_{i3}, \dots, v_{id}$  persamaan (1) merupakan proses update kecepatan setiap partikel dan persamaan (2) merupakan update posisi masing-masing partikel.

$$v_{ij}^{n+1} = w \cdot v_{ij}^n + C_1 r_1 (p_{ij}^n - x_{ij}^n) + C_2 r_2 (p_{gj}^n - x_{ij}^n) \quad (1)$$

Dimana:

$v_{ij}^{n+1}$  : kecepatan partikel iterasi selanjutnya

$v_{ij}^n$  : kecepatan partikel ke saat ini (iterasi ke  $n$ )

$w$  : bobot inersia

$C_1, C_2$  : koefisien akselerasi

$r_1, r_2$  : bilangan acak antara 0 dan 1

$p_{ij}^n$  : posisi partikel terbaik iterasi ke  $n$

$$x_{ij}^{n+1} = x_{ij}^n + v_{ij}^{n+1} \quad X = 1, 2, 3, \dots, d \quad (2)$$

Dimana :

$P_i^n$  merupakan posisi vector ke- $i$  terbaik yang didapatkan sampai iterasi ke- $n$  iterasi sedangkan  $P_g^n$  merupakan posisi terbaik dari semua partikel (dalam populasi).

### 1.4 k-Nearest Neighbour

*K-Nearest Neighbor* (k-NN) termasuk kelompok *instance-based learning*. Algoritma ini juga merupakan salah satu teknik *lazy learning*. Algoritma *k-Nearest Neighbor* adalah sebuah metode untuk melakukan lasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. k-NN dilakukan dengan mencari kelompok k objek dalam data *training* yang paling dekat (mirip) dengan objek pada data baru atau data testing [12]. Data *training* akan dievaluasi dengan  $k$  tetangga (neighbor) terdekat. Data *training* diproyeksikan ke ruang berdimensi banyak, yang mana masing-masing dimensi menjelaskan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi data *training*. Sebuah titik pada ruang ini ditandai kelas  $c$ , jika kelas  $c$  merupakan klasifikasi yang paling banyak ditemui pada  $k$  buah tetangga terdekat titik tersebut [13].

Algoritma k-NN untuk lebih jelas melihat hubungan antara data mining. Penentuan nilai  $k$  terbaik tergantung pada data. Nilai  $k$  yang tinggi bisa mengurangi efek noise pada klasifikasi, tetapi membuat batasan antara setiap kelas menjadi kabur. Sedangkan penentuan nilai  $k=1$  belum tentu bisa menjawab permasalahan data mining dalam hal ini tingkat validitas. Nilai  $k$  terbaik dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan *k-fold cross validation*. Untuk membedakan nilai  $k$  pada *cross validation* dengan nilai  $k$  pada k-NN, maka digunakan *n-fold cross validation* untuk mengacu kepada istilah yang sama yaitu *k-fold cross validation*.

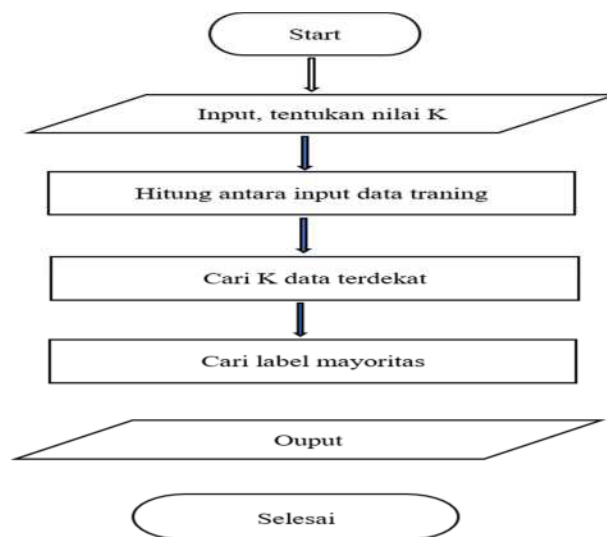
Pada *n-fold cross validation*, data dibagi sejumlah  $n$  dan data ke- $n$  digunakan sebagai data testing sedangkan data selain data ke- $n$  sebagai data pembentuk pola atau data training. Pada proses fold ke- $n$ , Parameter nilai  $k$  tertentu digunakan untuk menguji validitas data terhadap data testing menggunakan algoritma k-NN. Pada akhir proses, rata-rata error klasifikasi data dari masing-masing nilai  $k$  untuk seluruh *fold* dihitung. Dari proses tersebut diperoleh nilai  $k$  terbaik yang dapat dipakai sebagai  $k$  default dari algortima k-NN. k-NN efektif

bila melibatkan data training dalam jumlah yang besar dan bisa mengurangi efek *noise*. Tetapi k-NN juga memiliki beberapa kelemahan dalam hal kebutuhan memori yang besar dalam menghitung jarak antar *record*, penentuan nilai k secara *manual* dan kecepatan dalam mengklasifikasi data tergantung dari banyaknya data.

Secara umum untuk mendefinisikan jarak antara dua objek x dan y, digunakan rumus jarak Euclidean, seperti terlihat pada persamaan di bawah ini :

$$D_{xy} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3)$$

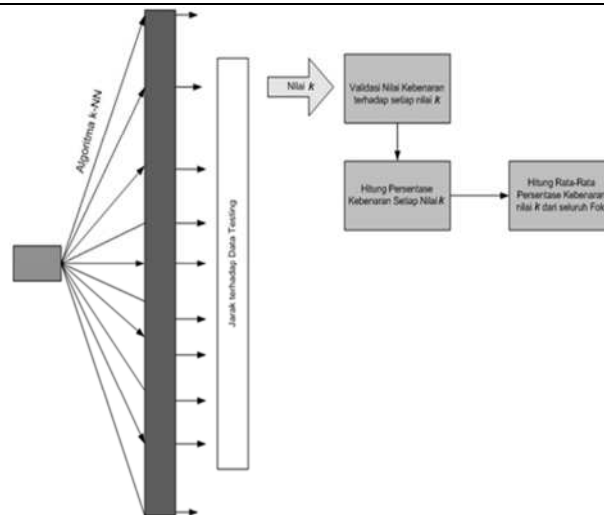
Jarak antara objek x dan y didefinisikan sebagai  $D_{xy}$ , dimana  $x_i$  merupakan record yang akan diprediksi dan  $y_i$  merupakan record data pola sedangkan nilai n didefinisikan sebagai jumlah atribut. Nilai i merujuk kepada *record* ke-i. Persamaan (2) merupakan rumus jarak untuk tipe data kontinu sehingga data nominal harus dikonversi ke bentuk numerik. Proses penghitungan jarak dapat digambarkan dalam tahapan algoritma k-NN. Berikut diagram alir k-NN



**Gambar 1: Gambar Alir Metode k-NN**

### 1.5 k-Fold Cross Validation

*n-fold cross validation* atau *k-fold cross validation* merupakan salah satu metode yang digunakan untuk mengetahui rata-rata keberhasilan dari suatu sistem dengan cara melakukan perulangan dengan mengacak atribut masukan sehingga sistem tersebut teruji untuk beberapa atribut input yang acak. *n-fold cross validation* diawali dengan membagi data sejumlah *n-fold* yang diinginkan. Dalam proses *cross validation* data akan dibagi dalam n buah partisi dengan ukuran yang sama  $D_1, D_2, D_3, \dots, D_n$  selanjutnya proses testing dan training dilakukan sebanyak n kali. Dalam iterasi ke-*i* partisi  $D_i$  akan menjadi data testing dan sisanya akan menjadi data training. Untuk penggunaan jumlah fold terbaik untuk uji validitas, dianjurkan menggunakan *10-fold cross validation* dalam model.



**Gambar 2: Contoh iterasi data dengan 10-fold cross validation**

Gambar 2 di atas memperlihatkan tahapan proses 10-fold cross validation dengan menggunakan model algoritma k-NN. Hitung jarak dari setiap data testing terhadap data training. Input parameter nilai k tertentu. Verifikasi hasil klasifikasi setiap nilai k dengan nilai klasifikasi sebenarnya dari data testing. Pada proses akhir dilakukan perhitungan rata-rata tingkat kebenaran atau tingkat error dari tiap fold ke-n terhadap setiap nilai k.

### 1.6 Root Mean Square Error

Root Mean Square Error (RMSE) merupakan parameter yang digunakan untuk mengevaluasi nilai hasil dari pengamatan atau pengukuran terhadap nilai sebenarnya atau nilai yang dianggap benar. RMSE ini dihitung pada saat transformasi koordinat selesai dilakukan. Dengan menggunakan RMSE, error yang ada menunjukkan seberapa besar perbedaan hasil estimasi dengan hasil yang akan diestimasi. RMSE digunakan untuk mencari nilai keakurasian hasil prediksi dengan data histori. Semakin kecil hasil yang dihasilkan semakin bagus pula hasil prediksi yang dilakukan. Secara umum, persamaan untuk menghitung besarnya RMSE dalam bidang dua dimensional adalah sebagai berikut :

$$RMSE = \sqrt{\frac{\sum (y_t - \hat{y}_t)^2}{n}} \quad (4)$$

Dimana :

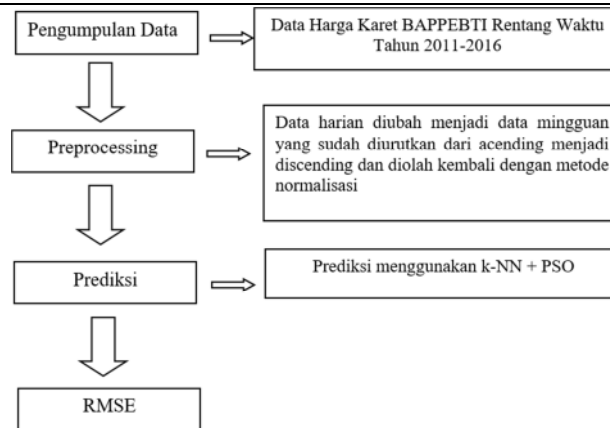
**RMSE** = Root Mean Square Error

**n** = Jumlah Sampel

**y<sub>t</sub>** = Nilai Aktual Indeks **y**

**t** = Nilai Prediksi Indeks

Adapun Jenis penelitian yang dilakukan adalah dengan metode eksperimen, yaitu untuk mengetahui pengaruh metode optimasi *Particle swarm optimization* pada *k-Nearest Neighbor* dalam menghasilkan nilai RMSE yang lebih baik, dengan melakukan pengujian pada sebuah data kemudian diimplementasikan *dirapid miner* dan membandingkan data dengan atau tanpa menggunakan optimasi.



**Gambar 3: Diagram Alur Metode**

Data preprocessing yang dilakukan adalah data yang berupa univariate ubah menjadi data multivariate agar menghasilkan nilai yang bisa diolah dan normalisasi data.

Normalisasi dibutuhkan sebelum proses data mining, agar tidak ada parameter yang mendominasi dalam perhitungan jarak antar data. Semua variabel akan dinormalisasi pada rentang 0 sampai dengan 1 [12] dengan menggunakan persamaan pada rumus berikut ini:

$$x' = \frac{x - \min value}{\max value - \min value} \quad (5)$$

Setelah data yang univariate dirubah menjadi data multivariate dan dinormalisasi langkah selanjutnya yang dilakukan adalah dengan mencari model terbaik untuk nilai terdekat dalam *k-Nearest Neighbour*. Kemudian setelah mendapatkan nilai tetangga terbaik dari *k-Nearest Neighbour* dilakukan pembobotan dengan menggunakan *Particle Swarm Optimization* dengan harapan agar didapatkan nilai RMSE untuk prediksi harga komoditi karet.

## 2. HASIL dan PEMBAHASAN

Dataset dalam penelitian ini menggunakan 288 record dan 5 atribut prediktor yang sudah dinormalisasi. Adapun metode yang digunakan untuk pengujian data ini adalah *k-Nearest Neighbor* dan dataset yang telah diubah menjadi data *multivariate* yang telah dinormalisasi. Data dinormalisasi dan dibagi menjadi beberapa periode yaitu 1 periode hingga 5 periode. Dari kelima periode, dapat dipilih berdasarkan model terbaik menggunakan *k-NN*. Dalam melakukan prediksi, metode *k-Nearest Neighbor* memiliki cara yaitu dengan mencari tetangga terdekat dari angka yang menjadi label (*xt*).

Berikut rangkuman hasil uji coba arsitektur terbaik untuk setiap periode :

**Tabel 1: Nilai RMSE setiap periode**

Periode	K	Nilai RMSE
Pertama	13	0.098
Kedua	15	0,91
Ketiga	11	0,089
Keempat	15	0,089
Kelima	15	0,087

Berdasarkan tabel 1 diatas didapatkan model arsitektur k-NN terbaik dengan nilai RMSE terendah. Arsitektur k-NN pada uji coba tersebut adalah jumlah K=5 dengan nilai RMSE terendah sebesar 0,87 didapatkan pada data 5 variabel periode.

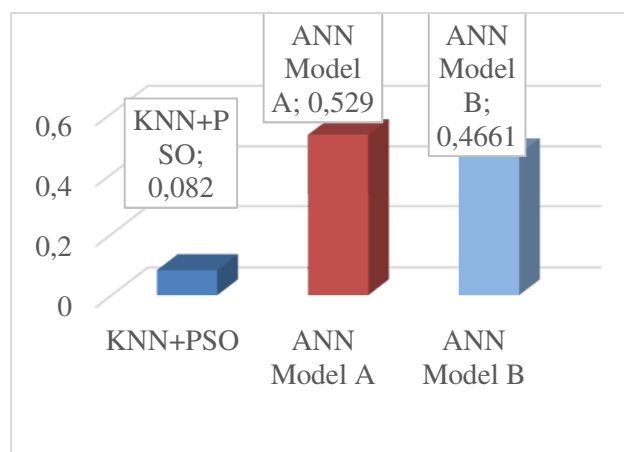
Setelah melakukan percobaan untuk mencari model terbaik dengan metode k-NN, maka didapatkan hasil terbaik pada periode ke 5 dan jumlah K = 15 menghasilkan nilai RMSE yang rendah yaitu 0.087.

Selanjutnya pengujian dengan menambahkan metode PSO untuk pembobotan dapat dilihat pada tabel 2 di bawah ini :

**Tabel 2 : Hasil Pengujian Nilai *Global Best Weight***

<i>Population Size</i>	<i>Max of Generation</i>	<i>Inersia Weight</i>	<i>Local Best Weight</i>	<i>Global Best Weight</i>	RMSE
15	100	0,3	0,3	0,1	0,083
15	100	0,3	0,3	0,2	0,083
15	100	0,3	0,3	0,3	0,082
15	100	0,3	0,3	0,4	0,083
15	100	0,3	0,3	0,5	0,083
15	100	0,3	0,3	0,6	0,083
15	100	0,3	0,3	0,7	0,083
15	100	0,3	0,3	0,8	0,082
15	100	0,3	0,3	0,9	0,083
15	100	0,3	0,3	1	0,082

Berdasarkan tabel 2 di atas didapatkan hasil parameter terbaik untuk pembobotan dengan menggunakan *Particle Swarm Optimization* adalah nilai populasi = 15, nilai *max of generation* = 100, nilai bobot *inersia* = 0.3, nilai bobot *local best* = 0.3 dan nilai *global best* = 0.3 dengan nilai RMSE terendah yaitu 0.082.



**Gambar 4: Perbandingan Nilai RMSE kNN+PSO dan ANN**

Berdasarkan dari grafik tersebut diatas maka dilihat bahwa nilai RMSE yang diperoleh dari percobaan menggunakan metode k-NN dan PSO sebagai optimasi menghasilkan RMSE yang lebih rendah yaitu 0,082 dibandingkan dengan percobaan menggunakan ANN dengan hasil RMSE model A 0,529 dan Model B 0,461.



Setelah melakukan pengujian untuk mencari model terbaik dari k-NN yang dioptimasi dengan PSO, tahap selanjutnya adalah menerapkan model terbaik dari uji coba untuk melakukan prediksi. Data yang digunakan untuk melakukan prediksi adalah dataset harga karet sebanyak 5 atribut untuk 4 minggu kedepan. Hasil prediksi dapat dilihat dalam tabel 4.36 di bawah ini.

**Tabel 3 : Hasil Prediksi untuk 4 Minggu ke Depan**

Tahun	Minggu ke-	Hasil Prediksi Harga
2016	289	0.24827
2016	290	0.26725
2016	291	0.26205
2016	292	0.25597

Setelah didapatkan hasil prediksi seperti pada tabel 3 diatas kemudian dilakukan denormalisasi. Denormalisasi dilakukan untuk mengembalikan ukuran data yang telah dinormalisasi sebelumnya untuk mendapatkan nilai dari data yang asli. Denormalisasi dilakukan pada hasil keluaran dari *testing* berupa prediksi harga karet. Adapun persamaan untuk denormalisasi [25] adalah sebagai berikut :

$$\text{Denormalisasi} = Y (\text{max} - \text{min}) + \text{min}(5)$$

Dimana :

Y = Hasil keluaran dari *testing*

Min = Data minimum

Max = Data maksimum

$$\text{Denormalisasi} = Y (\text{max} - \text{min}) + \text{min}$$

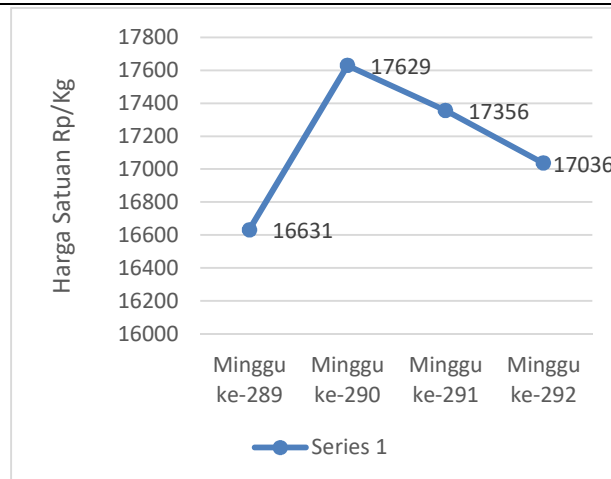
$$= 0.248266(56174 - 3572) + 3572$$

$$= 16,631$$

Kemudian dengan cara yang sama maka dihasilkan denormalisasi hasil prediksi harga seperti dalam tabel 4 berikut :

**Tabel 4: Hasil Prediksi pada 4 Minggu kedepan setelah Denormalisasi**

Tahun	Minggu ke-	Hasil Prediksi Harga
2016	289	16.631
2016	290	17.629
2016	291	17.356
2016	292	17.036

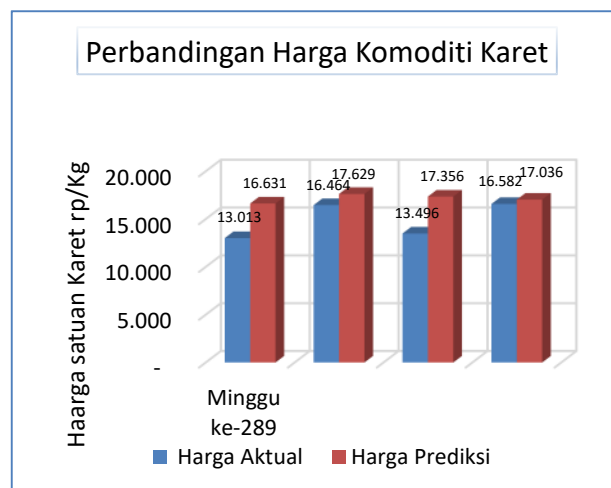


**Gambar 5: Grafik Hasil Prediksi Harga Karet**

Langkah selanjutnya adalah membandingkan nilai aktual Karet pada BAPPEBTI dan nilai prediksi pada minggu ke 289-292 pada 2016 dapat dilihat dalam tabel 4.37 di bawah ini:

**Tabel 5 : Perbandingan Harga Prediksi**

Tahun	Minggu ke-	Harga Aktual	Harga Prediksi
2016	289	13,013	16,631
2016	290	16,464	17,629
2016	291	13,496	17,356
2016	292	16,582	17,036



**Gambar 6 : Grafik Perbandingan Harga Komuditi Karet**

### 3. KESIMPULAN

Peramalan harga komoditi karet dengan menggunakan algoritma *k-Nearest Neighbor* dan *Particle Swarm Optimization* yang sudah dilakukan. Berdasarkan hasil eksperimen yang telah peneliti lakukan maka dapat disimpulkan bahwa

Algoritma *k-Nearest Neighbor* berbasis *Particle Swarm Optimization* lebih baik dibandingkan dengan algoritma *Neural Network* yang memakai dua model jaringan saraf tiruan pada penelitian terdahulu yang masih menghasilkan nilai error terlalu besar. *K-Nearest Neighbor* (*k-NN*) berbasis optimasi *Particle Swarm Optimization* (PSO) menghasilkan nilai RMSE sebesar 0.082 yang lebih rendah di bandingkan memakai metode *Neural Network* yang memakai dua model . Model A RMSE 0,5295 dan MAPE 1,2498 sedangkan Model B RMSE 0,4661 dan MAPE 0,9961.

### DAFTAR PUSTAKA

- [1] Direktorat Perkebunan, “Rencana Strategis,” 2015.
- [2] I. Santoso, U. Effendi, and C. Fauziya, “Penerapan Jaringan Syaraf Tiruan untuk Peramalan Permintaan Komoditas Karet di PT. Perkebunan Nusantara XII Surabaya,” vol. 8, no. 1, pp. 46–54, 2007.
- [3] Kristiyanti, Dinar Ajeng, “*Analisis Sentimen Review Produk Kosmetik menggunakan Algoritma Support Vector Machine dan Particle Swarm Optimization sebagai Metode Seleksi Fitur*, Seminar Nasional Inovasi dan Tren (SNIT)”, 2015
- [4] Alkhatib, Khalid, et al. *Stock Price Prediction using K-Nearest Neighbor (k-NN) Algorithm*. International Journal of Business, Humanity and Technology. Vol. 3 No 3, 2013
- [5] Bahri, Raden Sofian dan Irfan Maliki. *Perbandingan Algoritma Template Matching dan Feature Extraction pada Optical Character Recognition*. Jurnal Komputer dan Informatika (KOMPUTA), Edisi I Volume 1, Maret 2012.
- [6] Indriyanto, Jatmiko, Purwanto dan Catur Supriyanto. *Algoritma K-Nearest Neighbor berbasis Chi-Squared untuk Prediksi Nasabah Asuransi*. Universitas Dian Nuswantoro, 2014
- [7] Kristiyanti, Dinar Ajeng. *Analisis Sentimen Review Produk Kosmetik menggunakan Algoritma Support Vector Machine dan Particle Swarm Optimization sebagai Metode Seleksi Fitur*. Seminar Nasional Inovasi dan Tren (SNIT), 2015
- [8] Sumarno, “Aplikasi *Wavelet Recurrent Neural Network* Untuk Prediksi *Data Time Series*”, Matematika ITS, Surabaya, 2009.
- [9] Sukmawan Nathan, “*Prediksi Indeks harga Saham Menggunakan Kombinasi Algoritma Particle Swarm Optimization (PSO) dan Time Variant Fuzzy Time Series (TVFTS)*”, Bandung
- [10] Makridakis, dkk. “*Metode dan Aplikasi Peramalan*”. (Edisi ke-2). (Terjemahan Untung S.A. dan Abdul Basith). Jakarta: Erlangga, 1995
- [11] Zaitun Time Series Developer Team (2009), “*Aplikasi Untuk Analisis Statistik, Pemodelan dan Peramalan Data Time Series*”,
- [12] Ndaumanu, Ricky Emanuel, dkk. *Analisis Prediksi Tingkat Pengunduran Diri Mahasiswa dengan Metode K-Nearest Neighbor*. Jatisi, Vol.1 No.1, 2014
- [13] Nanja, Muis dan Purwanto. *Metode K-Nearest Neighbor berbasis Forward Selection untuk Prediksi Harga Komoditi Lada*. Jurnal Pseudocode, Volume 2 Nomor 1. ISSN : 2355-5920, 2015